

instant3Dhub & WEBVIS

Installation, Configuration & Operation Manual

V1.6, LAST UPDATE: 18/12/2016

Table of Contents

1	Installation and Start	3
1.1	Linux	3
1.1.1	Installation	3
1.1.2	Installation locations.....	3
1.1.3	Starting and stopping the Service	4
1.1.3.1	SystemV Init	4
1.1.3.2	SystemD Init	5
1.1.4	instant3Dhub Configuration	6
1.1.4.1	InstantHub Parameter.....	7
1.1.4.2	InstantService Parameter	8
1.1.4.3	instant3Dhub Log Levels.....	9
1.1.4.4	instant3Dhub URN resolver configuration.....	9
1.1.5	Firewall configuration	10
1.1.5.1	IP tables.....	11
1.1.5.2	Firewalld.....	11
2	Run WebVis	12
2.1	Locating instant3DHub	12
2.2	Creating an empty webVis Application.....	13
3	Appendix A: Installation Example	14
4	ANNEX B: Common Tasks	16
4.1	License management	16
4.1.1	License Generation.....	16
4.1.2	License Installation	16
4.1.2.1	License Installation	16
4.1.2.2	License Verification.....	17
4.2	Cache Relocation	18
4.3	Changing Users.....	19
4.4	Installing instantHub / instantService on separate machines.....	20
4.5	Installation behind a proxy	21
4.5.1	Proxy example	22
4.5.1.1	General configuration	22
4.5.1.2	Configuration.....	22

1 Installation and Start

1.1 Linux

1.1.1 Installation

1. Obtain the latest InstantReality Dailybuild, for example from <ftp://ftp.igd.fraunhofer.de/outgoing/irbuild>
2. Use the distributions packet management system to install the downloaded package and any missing dependencies.

An exemplary walkthrough with the most common customizations if given in Annex A.

1.1.2 Installation locations

<code>/opt/instantReality</code>	main installation location
<code>/opt/instantReality.custom</code>	location of custom configuration files
<code>/usr/local/bin</code>	application symbolic links
<code>/var/cache/instant3DHub</code>	cache location
<code>/var/log/instant3DHub</code>	log file location

1.1.3 Starting and stopping the Service

Starting and stopping the service is done via standard mechanisms. Either SystemV or SystemD can be used.

1.1.3.1 SystemV Init

1.1.3.1.1 Manual Service Start

Use the installed auto start scripts to manually start the web based services in the following order:

1. `/etc/init.d/InstantHubDB start`
2. `/etc/init.d/InstantHub start`
3. `/etc/init.d/InstantService start`
4. `/etc/init.d/InstantMeasurementService start`
5. `/etc/init.d/InstantDeliveryNginx start` (if available)

The service status can be queried using the status parameter (for example `/etc/init.d/InstantHub status`).

1.1.3.1.2 Manual Service Shutdown

Use the installed auto start scripts to manually shutdown the web based services in the following order:

1. `/etc/init.d/InstantDeliveryNginx stop` (if available)
2. `/etc/init.d/InstantMeasurementService stop`
3. `/etc/init.d/InstantService stop`
4. `/etc/init.d/InstantHub stop`
5. `/etc/init.d/InstantHubDB stop`

1.1.3.1.3 Automatic Startup/Shutdown

To install the auto start scripts for their corresponding run-levels (default 3 and 5) use **chkconfig --add <service name>** (for example `chkconfig --add InstantHub`). After the next reboot the services should be available, alternatively manually start the services without rebooting.

The auto start scripts can be removed from their corresponding run-level by using **chkconfig --del <service name>** (for example `chkconfig --del InstantHub`).

1.1.3.2 SystemD Init

1.1.3.2.1 Manual Service Start

Use **systemctl** to manually start the web based services in the following order:

1. systemctl start instantHubDB.service
2. systemctl start instantHub.service
3. systemctl start instantServiceTranscoder.service
4. systemctl start instantServiceMeasurement.service
5. systemctl start instantDeliveryNginx.service (if available)

The service status can be queried using the systemctl status command (for example systemctl status InstantHub.service).

1.1.3.2.2 Manual Service Shutdown

Use **systemctl** to manually shutdown the web based services in the following order:

1. systemctl stop instantDeliveryNginx.service (if available)
2. systemctl stop instantServiceMeasurement.service
3. systemctl stop instantServiceTranscoder.service
4. systemctl stop instantHub.service
5. systemctl stop instantHubDB.service

1.1.3.2.3 Automatic Startup/Shutdown

To install the auto start services for their corresponding run-levels (default multi-user.target and graphical.target) use **systemctl enable <service name>** (for example "systemctl enable instantHub.service"). After the next reboot the services should be available, alternatively manually start the services without rebooting.

The auto start services can be removed from their corresponding run-level by using **systemctl disable <service name>** (for example "systemctl disable instantHub.service").

1.1.4 instant3Dhub Configuration

The service configuration can be adjusted using their respective custom configuration files

<code>/opt/instantReality.custom/instantHub</code>	instantHub configuration
<code>/opt/instantReality.custom/instantService</code>	instantService configuration
<code>/opt/instantReality.custom/instantHubDB</code>	instantHubDB configuration
<code>/opt/instantReality.custom/<nginx conf></code>	nginx configuration Note: NGINX_CONF must be set inside <code>/opt/instantReality.custom/instantHub</code> to point to this file. On available options, please consult the nginx documentation directly
<code>/opt/instantReality.custome/instantHubDB.ini</code>	couchdb configuration Note: COUCHDB_ADDITIONAL_CONFIG_FILE must be set inside <code>/opt/instantReality.custom/instantHubDB</code> to point to this file. On available options, please consult the couchdb documentation directly.

1.1.4.1 InstantHub Parameter

HUB_LOG_LEVEL	--loglevel <level> sets the instantHub log level, for individual levels see section 1.1.4.3 default: --loglevel 1
HUB_DELAY	<time in seconds> Wait time between instantHubDB and instantHub start up
HUB_LOG_PATH	-l <log file directory> default: -l \${HUB_ROOT}/var/log/instant3DHub/hub
HUB_PORT	-p <server port number> default: -p 8080
HUB_ROOT	<install root dir> relative to which /opt/instantReality can be found, must be identical to SERVICE_ROOT default: /
HUB_SERVICE_GROUP	--group <zero conf group name> If multiple instantHub instances are used within same installation/subnet, their group names must be different default: --group InstantHub
HUB_SERVICE_RECONFIG	--allowReconfig <true false> If true enables clearCache from instantHub WebUI default: --allowReconfig true
HUB_SSL	-c <ssl certificate file> -k <ssl key file> default: not set
HUB_URN_RES_CFG	--urn-res-config <config dir> path to the directory where a custom urn resolver configuration can be found, for details see 1.1.4.4 default: not set
HUB_USER_SYS	The user under which the daemons run default: instantReality
HUB_TRANSCMD_CONFIG	--cmd-config <config file> path to a custom transcoder command configuration, for details see section Error! eference source not found. default: not set
HUB_TRANSCMD_PPIO_CONFIG	--cmd-prio-cfg <prio config file> path to a custom transcoder command priorities file, for details see section Error! eference source not found. default: not set

1.1.4.2 InstantService Parameter

SERVICE_CACHE_PATH	-c <cache directory> default: -c \${HUB_ROOT}/var/cache/instant3DHub
SERVICE_DELAY	<time in seconds> Wait time between instantHub and instantService start up default: 5
SERVICE_DELIVERY	--delivery_service=<delivery URL> Override complete delivery URL, if delivery host is not the same as the service host (e.g. for installations behind proxies or NAT setups). If set the delivery port settings below will be ignored. --replyorigin Use request server host name to build delivery URI, for NAT setups with no externally resolvable host names. Note: requires cache regeneration ! default: not used
SERVICE_DELIVERY_PORT	--delivery_port <delivery port number> 4712: built-in 4713: nginx, https/spdy if available 4714: nginx, http Note: requires cache regeneration ! --delivery_port 4714
SERVICE_LOG_LEVEL	--loglevel <level> sets the instantService log level, for individual levels see section 1.1.4.3 default: --loglevel 1
SERVICE_LOG_PATH	-l <log file directory> -l \${HUB_ROOT}/var/log/instant3DHub/service
SERVICE_PORT	-p <service port number> default: -p 4711
SERVICE_REG	-i <instantHub hostname>:<instantHub port> instantHub service registration host name and port default: -l localhost:8080
SERVICE_ROOT	<install root dir> relative to which /opt/instantReality can be found, must be identical to HUB_ROOT default: /
SERVICE_SSL	-s --delivery-cert <ssl certificate file> --delivery-key <ssl key file> default: no set
SERVICE_TRANS_CONFIG	--config <config file> service transcoder whitelist configuration, for details see section Error! Reference source not found. default: no set
SERVICE_WORKERS	-t <number of worker threads> default: -t 30

1.1.4.3 instant3Dhub Log Levels

Numerical Value	Level Description
0	Minimal information
1	General information
2	Request Lines
3	Reply Line
4	Request Header
5	Reply Header
6	Request Full
7	Reply Full
8	Additional Debug information

Higher log levels include all lower log levels.

1.1.4.4 instant3Dhub URN resolver configuration

A simple mapping between URNs and URLs is defined in a json file containing an array of rules. Each element of the array defines a rule to resolve a specific URN. Adding another rule is done by simply adding another element to the array.

```
[
  {
    "namespace": "igd",
    "specifier": "i3dh:data",
    "urlTemplate": "http://gpucloud01.igd.fraunhofer.de/data/$(1)",
    "urlContentType": [ "openjt", "plmxml" ]
  }
]
```

In the example above, the URN `urn:igd:i3dh:data:73102` would be converted to <http://gpucloud01.igd.fraunhofer.de/data/73102>. Any parameters found after the pattern `urn:<namespace>:<specifier>` are expected to be separated by `:"`, and bound sequentially to the placeholders `$(1)`, `$(2)`, ... in the `urlTemplate`.

The field specifying content types is important, as these define which mime-types can be expected from the URL.

1.1.5 Firewall configuration

The following ports are used by the default Instant3DHub installation:

instantHub		
	tcp: 8080	instantHub webservice
	tcp: 8081	instantHub job submission for external tool chains
	udp: multicast	Zero conf protocol if applicable
instantService		
	tcp: 4710	https instantService webservice if applicable
	tcp: 4711	http instantService webservice
	tcp: 4712	Built-in http data delivery
	udp: multicast	Zero conf protocol If applicable
instantServiceMeasurement	tcp: 4730	https instantService webservice if applicable
	tcp: 4731	http instantService webService
	tcp: 62000 – 62023	Measurement rest service
nginx (if applicable)		
	tcp: 4713	https/spdy data delivery
	tcp: 4714	http data delivery
couchdb	tcp: 5984	http couchdb access
	tcp: 6984	https couchdb access if applicable

1.1.5.1 IP tables

A combination of the following rules have to be added manually to the servers' iptable configuration:

-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 8080 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4710 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4711 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4712 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4713 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4714 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 4730 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 5984 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--dport 6984 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp	-s <source host/net>	--match multiport --dports 62000-62023 -j ACCEPT

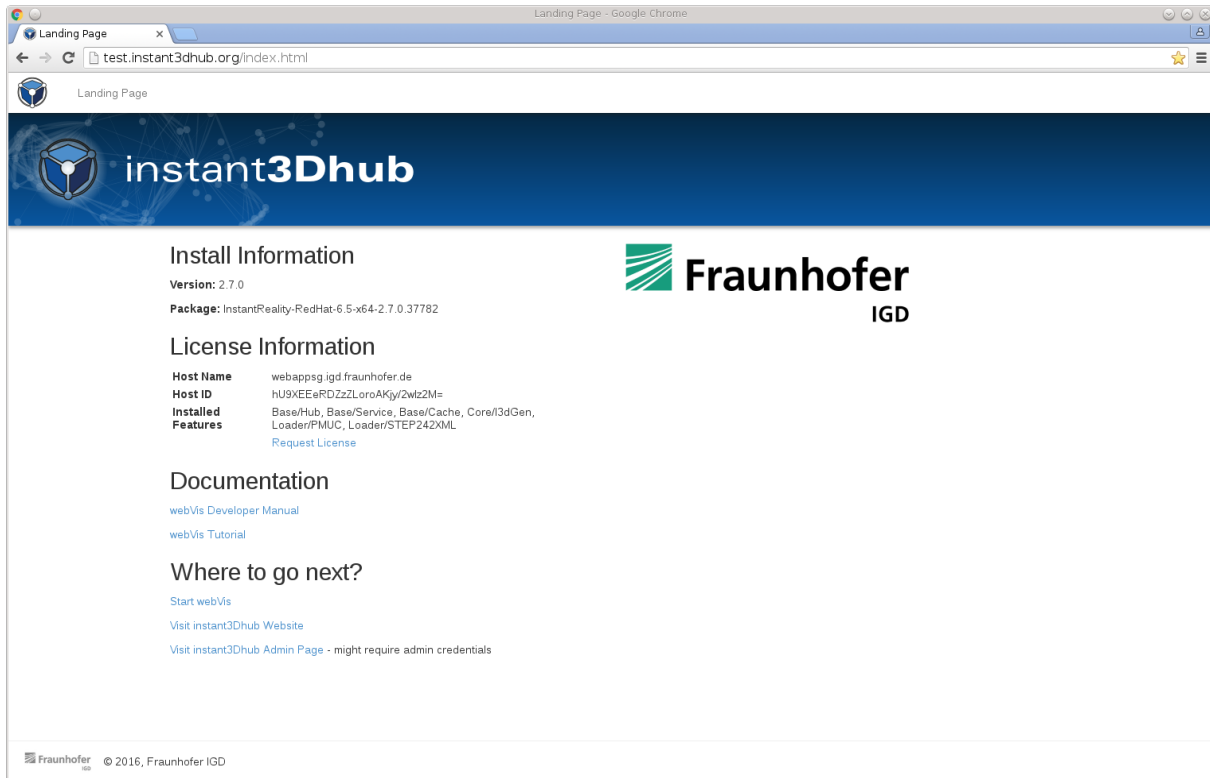
1.1.5.2 FirewallD

Use firewall-cmd to manage the **instant3Dhub** firewall ruleset, for example **firewall-cmd --add-service=instant3Dhub** to open the required ports. Adjustments as to which ports are opened can be made in `/usr/lib/firewalld/services/instant3Dhub.xml`.

2 Run WebVis

2.1 Locating instant3DHub

After a successful installation, you should be able to access the instant3Dhub (instant3Dhub landing page) installation using a web browser to connect to port 8080 on the instant3Dhub server (e.g. <http://<instant3Dhub host>:8080/index.html>).



The instant3Dhub landing page.

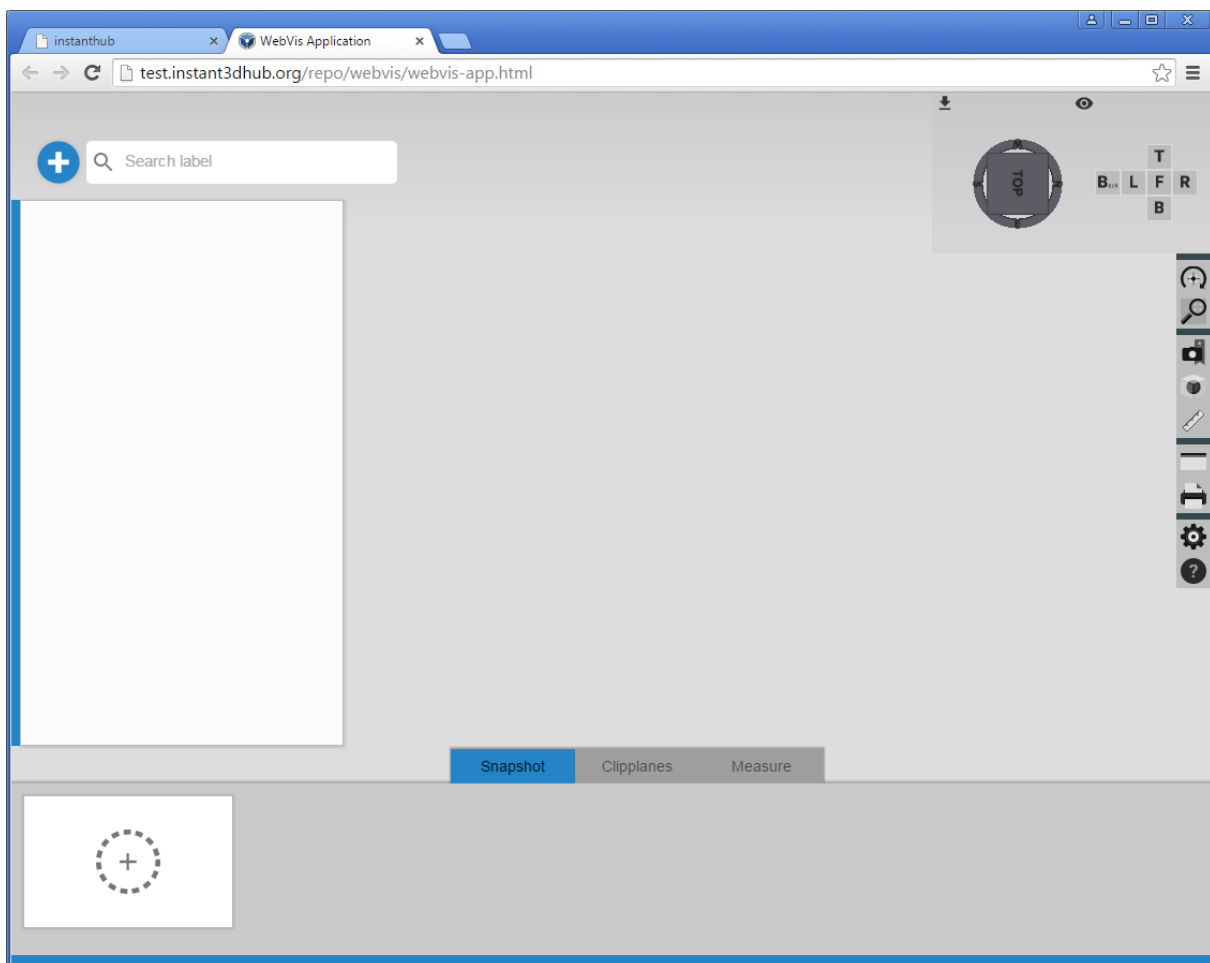
2.2 Creating an empty webVis Application

First, we create an empty webVis application using the following skeleton:

```
<html>
  <head>
    <title>My first webVis page</title>
    <script src="http://<instant3Dhub server>:8080/repo/webvis/webvis.js"></script>
  </head>
  <body>
    <webvis-full></webvis-full>
  </body>
</html>
```

<instant3Dhub server> must be replaced by the actual machine name / ip-address.

The example html code will create an empty webVis viewer as shown below.



An empty webVis application

3 Appendix A: Installation Example

The following illustrates a sample installation procedure including the most common customizations

1. Obtain the latest InstantReality software archive, for example from <ftp://ftp.igd.fraunhofer.de/outgoing/irbuild>
2. Use the distributions packet management system to install the downloaded package and any missing dependencies.
3. Request, install license:
Please refer to Annex B section 4.1 for details
4. If required, change the user used to run the instant3Dhub framework services:
Please refer to Annex B section 4.3 for details
5. If required, change the location of the instantService cache location:
Please refer to Annex B section 4.2 for details
6. Within `/opt/instantReality.custom/instantService` set the number of service workers using the `SERVICE_WORKERS` variable.
If the complete instantService framework is deployed on a single system the recommended value is `<num-cpus> - 2` in order to leave some resources for the instantHub and instantDelivery processes.

Example: `export SERVICE_WORKERS="--t 8"`

7. If the systems hostname is not configured to provide the fully qualified domain name, in order to ensure stable operation, the delivery service must be explicitly configured to use the FQDN instead of the hostname (default).

In `/opt/instantReality.custom/instantService` set the `SERVICE_DELIVERY` variable accordingly.

Example:

```
export SERVICE_DELIVERY="--delivery_service http://<fqdn>:4714"
```

8. If ssl is required to secure connections between endpoints execute the following steps to enable ssl support:

8.1. **instantHub:** to enable SSL for the instantHub instance, use the `HUB_SSL` variable in `/opt/instantReality.custom/instantHub` to configure the key and certificate files.

Example:

```
export HUB_SSL="-c <ssl certificate file> -k <ssl key file>"
```

8.2. **instantService:** to enable ssl for a particular instantService instance use the `SERVICE_SSL` variable in `/opt/instantReality.custom/instantService` to configure to the SSL certificate and key files as well as the fact that connections to the instantHub instance require ssl encryption (option `-s`)

Example Service:

```
export SERVICE_SSL="-s --delivery-cert <ssl certificate file> \  
--delivery-key <ssl key file>"
```

8.3. **instantDelivery (nginx):** Create a custom nginx configuration file by copying the default from `/opt/instantReality/opt/nginx/conf/nginx.conf` to `/opt/instantReality.custom/`.

Use the `NGINX_CONF` variable in `/opt/instantReality.custom/instantHub` to enable the new nginx configuration file.

Example:

```
export NGINX_CONF="-c /opt/instantReality.custom/nginx.conf"
```

Within the newly created nginx configuration file update the `ssl_certificate` and `ssl_certificate_key` variables in order to enable ssl support.

8.4. **instantService:** The delivery location in the instantService must be updated to the newly configured nginx location. The variable `SERVICE_DELIVERY` in `/opt/instantReality.custom/instantService` can be used:

Example:

```
export SERVICE_DELIVERY="--delivery_service \  
https://<fqdn>:<port>"
```

9. Start the services as described in section 1.1.3. and verify their operation.

4 ANNEX B: Common Tasks

4.1 License management

4.1.1 License Generation

In order to generate the license the host id of the system must be provided to the license issuer. After the installation of the InstantReality software package the host id can be retrieved by running the aopt command using the -L option, as shown below:

```
# aopt -L
License has no 'hub/aopt' feature!
WARNING Avalon   Unable to acquire license for feature hub/aopt
=====
! Unlicensed Version: limited functionality and only for non-commercial use !
=====

Command Line: aopt -L
LOG      Avalon   Read time: 0.000041
=====

Call: hostID with 0 param
HostId: +TGX8wFH8ksN2msKZkukWAqJaqY=
LOG      Avalon   Stage time for hostID: 0.000607
LOG      Avalon   Total run time for aopt: 0.074419
```

The string displayed after “HostId:” is the host identification string. To request a license the host id together with the host name has to be send to the license issuer.

4.1.2 License Installation

After providing the license issuer with the host id and host name, a license will be issued.

4.1.2.1 License Installation

On receiving the license, it is recommended to place the license file with the other InstantReality customizations in /opt/instantReality.custom. After placing the license the environment variable **TALIC_LICENSE_FILE** must be used to pass the location of the license file to the instantReality platform. The recommended method is to add the following line to the **/opt/instantReality.custom/instantHub** file:

```
export TALIC_LICENSE_FILE=/opt/instantReality.custom/<license file name>
```


4.1.2.2 License Verification

After installing the license the initial verification can be done using the aopt command. After running the following two commands:

```
export TALIC_LICENSE_FILE=/opt/instantReality.custom/<license file name>
aopt -L
```

The output should indicate a licensed version of aopt, in **bold** below:

```
# aopt -L
LOG      Avalon   Init: 27/471, V2.6.0 build: R-29951 Oct 14 2015
=====
License found: full functionality and commercial use granted.
=====

Command Line: aopt -L
LOG      Avalon   Read time: 0.000030
=====

Call: hostID with 0 param
HostId: +TGX8wFH8ksN2msKZkukWAqJaqY=
LOG      Avalon   Stage time for hostID: 0.000568
LOG      Avalon   Total run time for aopt: 0.154858
```

To verify that the instant3Dhub service picked up a valid license, please check the current instantService log file (default location `/var/log/instant3Dhub/service`). After restarting the instant3Dhub services the result of the license check will be shown at the end of the log file. If **success** is reported, the instant3Dhub platform will be fully operational.

4.2 Cache Relocation

In case the default cache location (`/var/cache/instant3Dhub`) is not suitable, the exact location can be adjusted to suit the operational needs. Two adjustments have to be made, first `instantService` has to be reconfigured to use the new location. This can be achieved, using the following parameter within a custom `/opt/instantReality.custom/instantService` file, by adding the following configuration variable:

Cache on disk location:

```
SERVICE_CACHE_PATH: -c <cache directory >
```

Second the `instantDelivery` nginx instance has to be configured to use the new location. In a custom nginx configuration file (see section **Error! Reference source not found.** on how to create and configure one) the root entry from location `/` has to be updated to reflect the new on disk location, as shown below:

From:

```
.  
.
location / {
    root    /var/cache/instant3DHub;
    .
    .
```

Change to:

```
.  
.
location / {
    root    /path/to/new/cache/location;
    .
    .
```

Note: After relocating an already filled cache, it has to be cleared and regenerated.

4.3 Changing Users

During installation the instant3Dhub framework sets the access rights for files and directories according to the default user instantReality. If the user is configured differently as part of the customization process by changing the `HUB_USER_SYS` variable in `/opt/instantReality.custom/instantHub`, care has to be taken to update the access right to files and directories accordingly. In particular for the log (`/var/log/instant3Dhub`, `/var/log/nginx`) and cache location (`/var/cache/instant3Dhub`) write access has to be granted to the user id used to run the instant3Dhub framework.

4.4 Installing instantHub / instantService on separate machines

The instantHub and instantService components do not necessarily need to run on the same machine. When installed on separate machines, the instantService must know where the instantHub is located in order to be able to register itself as a service provider. This can be done with the `SERVICE_REG` variable within `/opt/instantReality.custom/instantService` on the new machine.

```
export SERVICE_REG="-i <instantHub hostname>:<instantHub port>"
```

In addition, any proxy configurations referencing the original instantService location must be updated, and the instantService running on the instantHub server should not be started anymore.

4.5 Installation behind a proxy

The instant3DHub infrastructure can be run behind a reverse proxy. For this setup to work the following target URLs have to be forwarded by the proxy in the given order:

	URL	Target	Function
A	<proxy base>/<cache id>	<instantDelivery>/<cache Id>	retrieve cache entry
B	<proxy base>/service/<cache id>	<instantService:{4710 4711}>	query instantService
C	<proxy base>/couchDB	<instantHub:{5984 6984}>	query couchdb
D	<proxy base>/	<instantHub>/	webVis / instantHub UI access
E	<proxy base>/measure/(62[0-9]*)(.*)	<instantServiceMeasure>:\$1\$2	measurement service rest

Besides configuring the proxy rules the instant3DHub infrastructure has to be configured to generate and insert the appropriate URLs with respect to the proxy rules A, B, and C.

	parameter	value	configuration file / entry
A	--delivery_service	<proxy base>	/opt/instantReality.custom/instantService set: SERVICE_DELIVERY
B	--serviceproxy	<proxy base>/service/<cache id>	/opt/instantReality.custom/instantService add to : SERVICE_ADD_OPTIONS
C	--couchDBProxy	<proxy base>/couchDB	/opt/instantReality.custom/instantHub add to : HUB_ADD_OPTIONS

The <cache id> can be retrieved from /var/cache/instant3DHub/.uuid. Should the cache uuid be too large for the given proxy server (for example apache), it can be shortened by added --short-uuid to the SERVICE_ADD_OPTIONS parameter in /opt/instantReality.custom/instantService.

The following parameter, added to /opt/instantReality.custom/instantServiceMeasurementService (SERVICE_MEASURE_ADD_OPTIONS) could be used to shape the returned rest interface urls for the measurement service to match the proxy requirements.

	parameter	value	configuration file / entry
A	--rest_template	<proxy base>/measure/\${REST_PORT}	/opt/instantReality.custom/instantService set: SERVICE_DELIVERY

Note: After changing to the proxied location any existing old caches have to be cleared and regenerated.

4.5.1 Proxy example

4.5.1.1 General configuration

```
instant3DHub host           : i3dh.example.com
instant3DHub use ssl       : disabled

instant3DHub proxy         : proxy.example.com
instant3DHub proxy use ssl : disabled
instant3DHub proxy base path : /i3dh

cache uuid                 : 5165136b3a53228a-78f085d9-49a6-936a
```

4.5.1.2 Configuration

instant3DHub (i3dh.example.com)

/opt/instantReality.custom/instantHub

```
HUB_ADD_OPTIONS="${HUB_ADD_OPTIONS} --couchDBProxy https://proxy.example.com/i3dh/couchDB"
```

/opt/instantReality.custom/instantService

```
DELIVERY_SERVICE="--delivery_service https://proxy.example.com/i3dh"

SERVICE_ADD_OPTIONS="${SERVICE_ADD_OPTIONS} --short-uuid"
SERVICE_ADD_OPTIONS="${SERVICE_ADD_OPTIONS}
--serviceproxy https://proxy.example.com/i3dh/service/5165136b3a53228a-78f085d9-49a6-936a"
```

/opt/instantReality.custom/instantServiceMeasurementService

```
SERVICE_MEASURE_ADD_OPTIONS="${SERVICE_ADD_OPTIONS} --short-uuid"
SERVICE_MEASURE_ADD_OPTIONS="${SERVICE_ADD_OPTIONS}
--rest_template https://proxy.example.com/measure/\{REST_PORT}"
```

Apache proxy configuration (proxy.example.com)

```
RedirectMatch "^/i3dh$" "/i3dh/index.html"

ProxyPass /i3dh/5165136b3a53228a-78f085d9-49a6-936a
          http://i3dh.example.com:4713/5165136b3a53228a-78f085d9-49a6-936a
ProxyPassReverse /i3dh/5165136b3a53228a-78f085d9-49a6-936a
          http://i3dh.example.com:4713/5165136b3a53228a-78f085d9-49a6-936a

ProxyPass /i3dh/service/5165136b3a53228a-78f085d9-49a6-936a http://i3dh.example.com:4711
ProxyPassReverse /i3dh/service/5165136b3a53228a-78f085d9-49a6-936a http://i3dh.example.com:4711

ProxyPass /i3dh/couchDB http://i3dh.example.com:5984
ProxyPassReverse /i3dh/couchDB http://i3dh.example.com:5984

ProxyPass /i3dh/ http://i3dh.example.com:8080/
ProxyPassReverse /i3dh/ http://i3dh.example.com:8080/

RewriteRule ^/i3dh/measure/(62[0-9]*) (.*)$ http://i3dh.example.com:\$1\$2 [P,L]
```